

05/28/99
JC648 U.S.

BRUSSELS, BELGIUM
WARSAW, POLAND
HONG KONG, CHINA
BANGKOK, THAILAND
NEW YORK, NEW YORK
ATLANTA, GEORGIA

HUNTON & WILLIAMS

1900 K STREET, N.W.

WASHINGTON, D.C. 20006-1109

TELEPHONE (202) 955-1500

FACSIMILE (202) 778-2201

05/28/99
JC648 U.S. PTO
09/321743

RICHMOND, VIRGINIA
MCLEAN, VIRGINIA
NORFOLK, VIRGINIA
RALEIGH, NORTH CAROLINA
CHARLOTTE, NORTH CAROLINA
KNOXVILLE, TENNESSEE

BRIAN M. BUROKER
E-MAIL: BBUROKER@HUNTON.COM

FILE NO.: 53470.000012
DIRECT DIAL: (202) 955-1894

May 28, 1999

BOX PATENT APPLICATION

Assistant Commissioner for Patents
Washington, D.C. 20231

Re: **New U.S. Patent Application of**
Inventors: Lee C. HAROLD, Qing ZHU and Herbert LIEBL
Title: "System And Method For Network User Interface OLAP Report
Formatting"

Dear Sir:

Attached is a new application for filing in the United States Patent and Trademark Office, including thirty-seven (37) pages of specification, four (4) pages of claims, an Abstract, seven (7) pages of drawings (Figures 1-7) and an unexecuted Joint Declaration for Patent Application.

The filing fee is calculated as follows:

Basic Filing Fee:	\$760.00
Total Claims: 23	54.00
Independent Claims: 3	<u>0.00</u>
Total Filing Fee:	\$814.00

This application is being filed without a filing fee and without an executed Declaration under 37 C.F.R. § 1.53(d). Accordingly, applicants are believed to be entitled to a filing date based upon the U.S. Patent and Trademark Office's receipt of the attached specification and drawings. Therefore, it is respectfully requested that a filing date of May 28, 1999 be granted to this application. The filing fee and executed Declaration will be timely filed.

The undersigned's Deposit Account No. is 50-0206. A duplicate of this sheet is attached.

HUNTON & WILLIAMS

BOX PATENT APPLICATION

May 28, 1999

Page 2

Please direct all communication concerning this application to:

James G. Gatto, Esq.
Hunton & Williams
1900 K Street, N.W., Suite 1200
Washington, D.C. 20006-1109

Respectfully submitted,

By: Brian M. Buroker
Brian M. Buroker
Registration No. 39,125

BMB/smm

Enclosures

SYSTEM AND METHOD FOR NETWORK USER INTERFACE OLAP

REPORT FORMATTING

Related Applications

This application is related by subject matter to co-pending applications entitled “System and Method For OLAP Report Generation With Spreadsheet Report Within the Network User Interface,” Attorney Docket No. 53470.000006 and “System and
5 Method for Asynchronous Control of Report Generation Using a Network Interface” Attorney Docket No. 53470.000005, both of which are filed herewith.

Field of the Invention

This invention relates to a system and method for generating, formatting, and styling OLAP reports in a network interface.

10 Background of the Invention

The ability to act quickly and decisively in today’s increasingly competitive marketplace is critical to the success of any organization. The volume of data that is available to organizations is rapidly increasing and frequently overwhelming. The availability of large volumes of data presents various challenges. One challenge is to
15 avoid inundating an individual with unnecessary information. Another challenge it to ensure all relevant information is available in a timely manner.

One known approach to addressing these and other challenges is known as data warehousing. Data warehouses and data marts are becoming important elements of many information delivery systems because they provide a central location where a
20 reconciled version of data extracted from a wide variety of operational systems may be stored. As used herein, a data warehouse should be understood to be an

informational database that stores shareable data from one or more operational databases of record, such as one or more transaction-based database systems. A data warehouse typically allows users to tap into a business's vast store of operational data to track and respond to business trends that facilitate forecasting and planning efforts.

5 A data mart may be considered to be a type of data warehouse that focuses on a particular business segment.

Decision support systems have been developed to efficiently retrieve selected information from data warehouses. One type of decision support system is known as an on-line analytical processing system ("OLAP"). In general, OLAP systems analyze
10 the data from a number of different perspectives and support complex analyses against large input data sets.

There are at least three different types of OLAP architectures - ROLAP, MOLAP, and HOLAP. ROLAP ("Relational On-Line Analytical Processing") systems are systems that use a dynamic server connected to a relational database
15 system. Multidimensional OLAP ("MOLAP") utilizes a proprietary multidimensional database ("MDDB") to provide OLAP analyses. The main premise of this architecture is that data must be stored multidimensionally to be viewed multidimensionally. A HOLAP ("Hybrid On-Line Analytical Processing") system is a hybrid of these two.

20 ROLAP is a multi-tier, client/server architecture comprising a presentation tier, an application logic tier and a relational database tier. The relational database tier stores data and connects to the application logic tier. The application logic tier comprises a ROLAP engine that executes multidimensional reports from multiple end

users. The ROLAP engine integrates with a variety of presentation layers, through which users perform OLAP analyses. The presentation layers enable users to provide requests to the ROLAP engine. The premise of ROLAP is that OLAP capabilities are best provided directly against a relational database, *e.g.*, the data warehouse.

5 In a ROLAP system, data from transaction-processing systems is loaded into a defined data model in the data warehouse. Database routines are run to aggregate the data, if required by the data model. Indices are then created to optimize query access times. End users submit multidimensional analyses to the ROLAP engine, which then dynamically transforms the requests into SQL execution plans. The SQL is submitted
10 to the relational database for processing, the relational query results are cross-tabulated, and a multidimensional result set is returned to the end user. ROLAP is a fully dynamic architecture capable of utilizing precalculated results when they are available, or dynamically generating results from atomic information when necessary.

The ROLAP architecture directly accesses data from data warehouses, and
15 therefore supports optimization techniques to meet batch window requirements and to provide fast response times. These optimization techniques typically include application-level table partitioning, aggregate inferencing, denormalization of data structures, multiple fact table joins, and the use of specific RDB optimizer tactics which vary with each particular brand of relational database.

20 MOLAP is a two-tier, client/server architecture. In this architecture, the MDDDB serves as both the database layer and the application logic layer. In the database layer, the MDDDB system is responsible for all data storage, access, and retrieval processes. In the application logic layer, the MDDDB is responsible for the

execution of all OLAP requests. The presentation layer integrates with the application logic layer and provides an interface through which the end users view and request OLAP analyses. The client/server architecture allows multiple users to access the multidimensional database.

5 Information from a variety of transaction-processing systems is loaded into the MDDDB System through a series of batch routines. Once this atomic data has been loaded into the MDDDB, the general approach is to perform a series of batch calculations to aggregate along the orthogonal dimensions and fill the MDDDB array structures. For example, revenue figures for all of the stores in a state would be added
10 together to fill the state level cells in the database. After the array structure in the database has been filled, indices are created and hashing algorithms are used to improve query access times.

Once this compilation process has been completed, the MDDDB is ready for use. Users request OLAP reports through the presentation layer, and the application
15 logic layer of the MDDDB retrieves the stored data.

The MOLAP architecture is a compilation-intensive architecture. It principally reads the precompiled data, and has limited capabilities to dynamically create aggregations or to calculate business metrics that have not been precalculated and stored.

20 The hybrid OLAP ("HOLAP") solution is a mix of MOLAP and relational architectures that support inquiries against summary and transaction data in an integrated fashion. The HOLAP approach enables a user to perform multidimensional analysis on data in the MDDDB. However, if the user reaches the bottom of the

multidimensional hierarchy and requires more detailed data, the HOLAP engine generates an SQL statement to retrieve the detailed data from the source relational database management system ("RDBMS") and returns it to the end user. HOLAP implementations rely on simple SQL statements to pull large quantities of data into
5 the mid-tier, multidimensional engine for processing. This constrains the range of inquiry and returns large, unrefined result sets that can overwhelm networks with limited bandwidth.

As described above, each of these types of OLAP systems are typically client-server systems. The OLAP engine resides on the server side and a module is typically
10 provided at a client-side to enable users to input queries and report requests to the OLAP engine. Current client-side modules are stand alone software modules that are loaded on client-side computer systems. One drawback of such systems is that a user must learn how to operate the client-side software module in order to initiate queries and generate reports.

15 Although various user interfaces have been developed to enable users to access the content of data warehouses through server systems, many such systems experience significant drawbacks. One system enables the user to access information using a network interface device (*e.g.*, a web browser). Such systems provide the advantage of allowing users to access the content of a data warehouse through existing web
20 browser applications residing on their desktop or other personal computer connected over a network to the server system. Such systems however suffer from various drawbacks.

In existing web-based user interfaces, a user inputs a query or report request through the web browser, the web browser initiates the report to the server, and waits until the report finishes before providing results back through the web browser to a user. While the web browser is waiting for the report, the web browser "locks up,"
5 thereby requiring that the user open another browser window if another site or page is desired to be opened. Accordingly, if the delay between the report initiation and completion is lengthy, some users may incorrectly assume that the delay is caused by an error, when in reality the delay may simply be caused by the complicated nature of the query. Users may become impatient and resubmit the report by selecting the
10 reload or refresh feature of the web browser. Current systems treat each such request separately and, therefore, each time the user resubmits a request, another query for the same information is submitted to the server. Because the reason for the delay in most circumstances is the size or complexity of the query, the server is then hit with processing multiple complex or large queries simultaneously when only one such
15 query or report is desired. Accordingly, these multiple reports tax server resources unnecessarily.

Another drawback this example illustrates is that current systems focus on requests. Accordingly, each request is treated as a different request that is submitted to the server for processing. Accordingly, identical requests initiated by different
20 users are submitted to the server system for processing, thus utilizing the server resources twice to generate the same report. These and other drawbacks exist with current web-based data warehouse/server system user interfaces.

Summary of the Invention

An object of the invention is to overcome these and other drawbacks in existing systems.

Another object of the invention is to provide a system and method for enabling
5 formatting of OLAP reports for presentation in a network user interface.

Another object of the present invention is to provide a system and method that enables selected of predefined auto-formats that may be applied to a selected report or series of reports.

Another object of the invention is to provide a system and method for
10 providing levels of formatting to be applied to multiple OLAP report outputs for presentation in a network user interface.

These and other objects of the present invention are realized according to various embodiments of the present invention. Accordingly, an embodiment of the present invention comprises a network-based system for enabling users connected
15 over the network to an OLAP system to select formatting options for reports requested for processing by the OLAP system. The system enables a user to select a workbook comprising one or more reports to be requested for processing by the OLAP system. The user may also select a single stand-alone report. The system then enables a user to select the format for each report in the workbook, for one or more reports in the
20 workbook or to a stand-alone report. The format specified may be multiple levels of format that are applied in a hierarchical manner including global formats that apply to all reports in a workbook, report formats which apply to specific reports in the workbooks, attribute/element/metric formats which apply to specific attributes,

elements, and metrics of a report, and value formats which apply to specific values within an element of a report. The formats specified may also comprise merged report formats that specify the merger of two or more reports, combined grid and graphs within a report and an autoformat that enables each recipient of the report to specify
5 the format in which that recipient views the report when the report is received.

Other objects and advantages of the present invention exist.

Brief Description of the Drawings

Fig. 1 is a schematic block diagram of a method for generating reports asynchronously in accordance with an embodiment of the invention.

10 Fig. 2 is a schematic block diagram of a method for generating, monitoring, and canceling reports asynchronously in accordance with another embodiment of the present invention.

Fig. 3 is a schematic illustration of a report list in accordance with one embodiment of the invention.

15 Fig. 4 is a schematic illustration of a status page in accordance with an embodiment of the invention.

Fig. 5 is a schematic block diagram of an asynchronous report requesting network system in accordance with an embodiment of the invention.

20 Fig. 6 is a schematic block diagram of system components in accordance with an embodiment of the invention.

Fig. 7 is a schematic block diagram of a network-based system in which the asynchronous report generation system may operate in accordance with one embodiment of the invention.

Detailed Description of the Preferred Embodiments

Accordingly, a method of operating a network-based OLAP system is provided. The system may comprise an Internet-based web browser interface that connects users over the Internet to a web server system that connects to the OLAP
5 server system for processing requested reports.

A method 200 may be provided according to an embodiment of the present invention, as depicted in Fig. 1. In this method, a user system providing a user interface module may be connected over a network to an OLAP server system. The first step in method 200 then involves step 202 whereby the user may request the
10 processing of a report to be run on the OLAP system through a web browser or other user interface system. The report may then be transmitted by the web browser or other user interface system to the server system such as a network module in communication with the web server.

After a report has been requested, in step 204, the system compares that report
15 with other pending or recently completed reports for the user submitting the report and for other users connected to the system to determine if the requested report is the same as a pending or processed report. In an embodiment, if a particular report has recently been completed or is already in the queue to be processed by the server system, the system does not create a new report to be processed. Rather, in step 207, the system
20 adds the user to the report that has already been requested so that when the previously requested report completes, then all users that requested that report receive the results. If the report has already completed, the system immediately returns the results to the

user as part of steps 215 and 226. If the report has not been requested, then in step 208, a report request is generated and transmitted to the OLAP processing system.

In step 226, the system returns control of the network interface, such as a web browser or other user interface module, to the user so that the user may perform other tasks. Specifically, the server system may direct the web server to transmit a signal to enable the user to use the web browser to perform other tasks, including submission of another report. The web server may also transmit a page that informs the user that the report that was requested was received and is scheduled for processing, such as a status page as discussed below.

Then, in step 220, the system notifies the user when a report has been completed so that the user may review the results of the report. By returning control of the web browser to the user upon receipt of the report request, the system frees the user to submit multiple reports for processing in an asynchronous manner. Additional details and features of the present invention may be understood with respect to Figs. 2-7 below.

Fig. 2 illustrates a more detailed embodiment of method 200 in accordance with one embodiment of the invention. Specifically, in step 204, the system determines whether that report is the same, substantially the same or similar to a report that has previously been requested. To do so, the system may store or have access to all, or a certain predetermined subset of all, reports forwarded to the system within a predetermined period of time (e.g., several hours, several days, or several weeks).

In an embodiment, the system, in step 204, may compare the name of the report with the name of other reports maintained in a cache that comprise all reports already pending and reports that have been completed but not removed from the cache. Also, the system may compare the template, filter, and user view of a request
5 to the template/filter/user view combination of other requested reports in the cache. If the template/filter/user view combination selected are identical, then the reports may be deemed to be the same.

Also, two reports may also be determined to be the same for purposes of avoiding generating a new report even if the templates or filters are not identical. For
10 example, two reports may be deemed to be substantially the same if the template /filter/user view combination is within a predetermined range of a template/filter combination of a previously run or pending report. For example, if a report for gross sales of running shoes for March, 1999 in the U.S. has been forwarded for processing, it may be determined that a request for gross sales of a particular brand of shoes for
15 March, 1999 in the U.S. is substantially the same or similar enough to be the same report. Other methods of determining similarity of reports to reduce the number of jobs to be processed by the system may also be used.

In step 208, if the report is different from other pending or recently completed reports, then the report is generated by placing the report in the processing queue in a
20 known manner. Additionally, the report may be added to a report list in step 210. The report list may comprise a list of all reports generated within a predetermined period of time and may include pending and completed reports. The report list may contain a report entry for each report containing the names or identifier(s) of system users that

have requested that report. The report entry for each report may also contain additional information if desired (e.g., complete request parameters, user id, requests GUID, electronic mail address, employee numbers, metadata user id, project name, report types (prompt or non-prompt), auto-prompt selections, report description, report name, filter name, template name, folder name, requested time, submitted time, completed time, filter details, status, and operators being applied (such as AND, OR, and NOT)). Whether the report is new or previously requested, the request is then added to the report entry for the report in step 212. The request may include user name and/or other identification, grid/graph selected, time requested, etc. to identify the request.

It may also be desirable to notify the user of the status of the report request so in step 214, a status page may be generated. Therefore, in step 214, a status page may be generated and presented to the user by the system over the network. The status page may identify the name of the report, date and time submitted, and/or any other additional information. An example of a status page is depicted in Fig. 4. The status page may be displayed as a separate window and may be accessed by the user through viewing a report list for that user as described below. At this point, the system may enable the user to request another report, view old reports or perform other types of actions.

The status page may also enable the user to cancel a particular request, as for example, illustrated in Fig. 4 and described below. The user may also be able to cancel a report from a user report list, as for example, the user report list shown in Fig.

3, such as by clicking on the report entry and selecting a cancel icon, pull-down menu option, or other mechanism for canceling a report.

By treating each report separately and asynchronously, a report may be canceled at any time. As each report may have multiple users that have been added to that report, however, in one embodiment, a report is not canceled from the queue at the system until all users that have been added to that report have canceled. If a user cancels the report, the user's name or id is removed from the report entry for that report and that canceled user then cannot access the report results unless they resubmit another request for the report. The details of this process are depicted in steps 216-226.

Specifically, in step 216, the system determines whether a report request has been canceled. If a report request has been canceled by a user, the system removes the request for that from the report list in step 218. After removing the user from the report entry for that report, the system, in step 220, determines whether any users remain in the report entry for that canceled report. If any user names remain, method 200 returns to step 216. If no other user names remain in the report entry, the system terminates the report and deletes it from the queue of reports being processed by the server system in step 222.

If, in step 216, a report request has not been canceled, the system determines whether any of the reports are complete in step 224. If a report is complete, in step 226 the system notifies all of the system users identified in the report list for that report that the report has been completed. Notification may be made using e-mail, facsimile, pager, through the web interface such as through the web browser or other

user interface module using push technology, etc. or any other mechanism as described below.

If the report generation is not complete, in step 224 the system returns to step 216 to determine whether any report request cancellations have been entered into the system. This continues until the report generation is complete and all systems users are notified that the report generation is complete for all reports on the system. The process may involve polling whereby steps 216 and 224 are performed on a predetermined schedule, such as every seven seconds or so. Other schedules may also be used.

Method 200 provides numerous advantages for generating reports. Receiving report requests and processing those requests asynchronously enables users to operate other applications while a report is being generated. Additionally, by focusing on the identity of the report rather than a request for that report, duplicative report generation is minimized. Therefore, if several users request the same report, only one report is generated.

Fig. 3 illustrates a user-specific report list 300 in accordance with an embodiment of the invention. Report list 300 may contain a report list for reports submitted for a specific user within a predetermined period of time. That time may be a particular log-in session or a discrete time interval for example. For each report, report list 300 may contain a "name" field 302 which identifies the name of the report being generated. A folder field 303 may be provided for indicating the folder or other category that the report may be organized. A "submitted" field 304 may be used to identify the date and/or time the report request was submitted. Other fields, for

example, a “details” field 306 may be used to provide additional information, such as, the status of the report. Fields 302-306 may be sorted according to various criteria (e.g., ascending or descending order) by, for example, clicking on a field using a conventional computer mouse. It is to be understood that report list 300 may be
5 sorted according to the value of a field (e.g., user, date, report, etc.). A status field may also be provided and contain information as described below with respect to Fig. 4. Further, various icons or other symbols may be used to indicate the status or type of report.

As Fig. 3 depicts, various types of reports may be processed according to the
10 present invention and these different report types may be displayed differently in report list 300. For example, an auto-prompt report may be provided with details of the prompting to show the filters, metrics, what information has been drilled on, what was selected in the prompt, other filter details, and other parameters selected for the report.

15 As discussed above, to notify the user that the request has been received and is being processed, a status page may be presented to the user. A status page may be used to display the progress of a report being generated and a user-specific report list may be viewed.

Fig. 4 illustrates a status page 400 in accordance with one embodiment of the
20 invention. Status page 400 provides a view displaying the current status of a report’s execution cycle. The execution cycle identifies the progress of a report generation (e.g., 10% complete, Error, complete with errors, pending, complete, sending request, generating SQL, submitting job request, request in queue, running report, retrieving

results, cross-tabbing data, formatting data, report complete, error running report, report canceled, m of n reports completed). Status page 400 may contain a graphical display 402 of the progress of the report generation on a percentage complete basis. Selectable icons may also be displayed within status page 400. For example, a report
5 list icon 404 may be displayed which enables a user to display a report list. A “refresh” icon 406 may be used to manually update the progress of the report generation. As described in further detail below, this operation may be automated and set by a user. Additionally, a “cancel request” icon may be displayed which permits a user to cancel the generation of a requested report.

10 A refresh status page rate may be set by a system user or by a system administrator. The system maintains a master list of requested reports and stores the list on a server. The master list may be refreshed when caches expire, a report is canceled, or a report is removed from a cache for another reason, etc.

A system administrator master report list may also be provided according to
15 the present invention and enabled through network output module 22. The administrator report list may be similar to user report list 300 but may additionally comprise entries for each report to identify the users that have requested that report. Further, the system administrator report list may comprise all reports run within a predetermined period of time and may be broken out project by project if desired. The
20 report list may provide a mechanism that enables a system administrator to delete entire reports, groups of reports, to delete a particular user or groups of users from a particular report or modify the order of reports in the queue. The administrator may thereby control reports submitted to the server system. In one embodiment, all such

reports may be stored in a cache and the master report list may contain all cached reports plus all pending reports.

The information in the history list may include the name of the request, the report display mode (graph or grid), the date and time the request was last submitted, the status of the request, the users who have submitted the request and their identifiers or keys, filter details for report requests, and workbook report names for workbook requests. The administrator may refresh the list, delete a request or view details about a request. The details may include the filters and templates requested.

According to another embodiment of the present invention, users may be able to submit requests for workbooks to be processed by the server system on data in the data warehouse. A workbook may comprise complex reports that generally comprise a plurality of reports and template/filter combinations that are collected to be generated as a large complex report. Therefore, as the term "report" as used herein should be understood to comprise a single template/filter combination run against data in the data warehouse or a collection of one or more "reports" in a format known as a "workbook."

Generally, to generate a workbook request, the user logs into the server system through username and password combinations. The user interface then presents the user with options for designing a new workbook or modifying an existing workbook. To add a new workbook or modify an existing one, the user identifies the workbook (e.g., by naming it), and then selects one or more predefined reports or template/filter combinations to be included in the workbook. The user also selects the sequence of the reports and template/filter combinations. The server system may present the

available reports and template/filter combinations by accessing the data warehouse, as for example, with a ROLAP engine by displaying the available table entries from the relational database structure in the data warehouse. Auto-prompted and drill-down reports may also be included in a workbook.

5 A workbook is processed as a collection of reports, with each report being processed and then when all reports in the workbook are complete, the workbook is formulated and the users requesting the workbook are then notified. The reports in a workbook may be processed asynchronously or sequentially depending on the types of reports.

10 According to an embodiment of the present invention, the user may specify the format of each report in the workbook or to a stand-alone report being requested. As such, predefined or user defined formats may be selected and have that format map to many formatting settings. For example, the user may pick the “elegant” auto-format, which may be defined by a combination of fonts, styles, colors, banding, spacing, etc.

15 The entire report or workbook for which that format is selected may then have the “elegant” auto-format applied. In this embodiment the system may enable the user to select between multiple levels of formats that are applied in a hierarchical fashion. For example, four levels of formats may be specified as follows.

 The highest level format may be a global format that applies to all reports in
20 the workbook. Global reporting provides the base level of formatting or a default for all reports that are not otherwise specified. Global formatting may be over-ridden by lower levels of formats. This formatting applied may be, for example, formatting used in GridX AutoFormat. Formats that may be selected as global formats include

an AutoFormat style that applies a format to all reports in the workbook, and server or client macros that run when the workbook is executed by the OLAP system. Server macros allow the user to manipulate the workbook report data before it is returned to the browser on the user system. After being run once on the OLAP server, the
5 workbook may be cached for all users of the system to access and manipulate the workbook through the use of server macros. Client macros may be launched to automatically manipulate the display of data already returned to the user interface system, such as a browser, to make additional functionality available. Global formats may also include provision of a worksheet for performance and error handling
10 including debug mode and log statistics.

The next lower level format may be a report level format. In this level, the formats specified apply only to a particular report and not to other reports in the workbook. This format may position a report within a workbook or worksheet and select an AutoFormat for a report.

15 Autoformat comprises predefined format styles by specifying parameters for an ActiveX type application, for example. The parameters include name, gridline, font, banding. Gridline defines which of the following gridline properties apply to the report: topbar, bottombar, leftbar, rightbar, topsep, leftsep, totalssep, hgrid, and vgrid. Font defines which of the following font properties apply: genfont, toplabel, leftlabel,
20 valuefont, totalslabel, totalvalue, leveltop, levelleft, and levelvalue. Banding defines which of the following banding properties are applied: level, width, darkcolor, lightcolor, and inlabel.

The topbar gridline property determines the content of the top gridline of the report. The bottombar gridline determines the content of the bottom gridline of the report. The leftbar gridline determines the content of the leftmost gridline of the report. the rightbar gridline determines the content of the rightmost gridline of the report. The topsep determines the content of the line that separates the title labels in the top header area of the report from the raw data. The leftsep determines the line that separates the title labels in the left header area of a report from the raw data. The totalsset determines the line that separates totals from other data values. The Hgrid determines how horizontal lines should be applied to the data area of the report. The Vgrid determines how vertical lines should be applied to the data area of a report.

The genfont attribute determines the font style of all cells in a report unless the cells are defined by one or more of the other font properties. The toplabel font is the font for the top label area (column headers). The leftlabel font is the font for the left label area (row headers). The valuefont is the font of the data value area. The totalslabel is the value of the label of the totals area of the report. The totalsvalue is the font for the totals value of the report. The leveltop is the font for the top labels at a specific level. The levelleft font defines the font for the left labels at a specific level. The levelvalue is the font of the data values at a specific row level.

For banding, the level defines the level to which banding should be applied. The width defines the number of banding rows or columns that should be displayed with the same color. The darkcolor defines the dark band background color. The lightcolor defines the light band background color. The inlabel defines whether or not banding extends into the headers in the label areas.

Through selecting values for these attributes, the autoformat applies to define the format of a report in the workbook.

Autoprompt reports are reports that prompt the user to input some of the parameters, such as the filter criteria, for example, for the report when it is initiated.

5 Positioning a report within a worksheet may include vertical stacking which specifies the distance (in rows, for example) that the selected report may be placed from the report preceding it and the distance (in columns, for example) from the left sheet margin. If no report is specified above the selected report, then the system uses the top sheet margin as the bottom edge from which to count rows. Also, horizontal
10 stacking may be provided which enables a user to specify the distance (in columns, for example) that the selected report should be placed from the report to the left of it and the distance (in rows, for example) from the top sheet margin. If no report is specified to the left of the selected report, then the system uses the left sheet margin as the edge from which to count columns. The system may also enable specification of the
15 absolute cell reference from which the top left corner of the report is to be placed.

The next lower level format may be an attribute/element/metric level format which overrides the global and report level formats for certain attributes, elements, and metrics within a particular report. The next lower level format may comprise the value level format that overrides the global, report, and attribute/element/metric level
20 format for a particular intersection of a metric and an element which establish a value.

Another level of formatting, a worksheet level formatting may also be applied. Worksheets may comprise collections of reports within the workbook and a worksheet

level format may be provided between the global and report level formats described above.

These formatting options for the workbooks may be selected by the user through a graphical user interface in the network user interface, such as the web browser. This process may be part of the request for a report to be processed by the OLAP system. Input may be through use of known techniques including direct entry into a field, pull-down menus, button selection, or other such techniques. Further, the formats may be from predefined styles that provide the parameters of the format. The user may also be able to define styles through a graphical user interface by saving formats input for a workbook or report as a style for later use with other workbooks or reports.

Also, formatting may be provided by the user through scripting. The user may use scripting, such as through the use of VBA or other scripting systems, to do completely programmatic and specialized formatting. This is possible because the data hooks (or "metadata") related to how formatting is applied are provided as parameters that may be used in scripts to modify formats. Through scripting, general and maintainable formatting may be provided rather than requiring hard-coded formatting within a grid or graph generated as a report.

Workbooks may be implemented through leveraging spreadsheet application programs, such as Microsoft Excel™. Through these spreadsheet applications, formatting of multiple reports into a combined workbook may be provided. Accordingly, the programming language in the spreadsheet application may be used to configure the format and arrangement of reports within the workbook. For example,

Visual Basic for Applications may be used in the Excel™ spreadsheet software application. This provides flexibility in formatting and combining reports to create new ones and in building an interactive spreadsheet application for the end user. For example, a spreadsheet application for a stock market workbook may provide pull-down menus for selecting different time ranges and industries which then update
5 instantly/dynamically. Because these programming languages and spreadsheet programs, such as VBA and Excel are open standards, they are widely used and therefore, the usefulness of the workbook feature is enhanced.

Also, using Excel™, for example, enables users to select various properties
10 and formatting for the reports that are to be generated. Integration of Excel™ with the invention allows users to use all of the functionality offered in Excel™ and to apply that functionality to the reports generated using the invention. For example, Excel™ offers existing macros that may be used to automate work flow or other features for the reports. Additionally, these macros may be modified or additional macros may be
15 recorded to perform a specific operations desired by a user. The macros may be used, for example, to facilitate work flow. Generating reports in Excel™ also enables reports to display grids and graphs of the same report at the same time, display headers and footers, deliver multiple worksheets in a single workbook, support logical exceptions through use of overwriting rules, allows administrators to create formatted
20 reports, etc.

A detailed embodiment of a system for enabling users to asynchronously generate report requests is depicted in Fig. 5. Such a system 50 comprises a data warehouse 12, server system 14, network output module 22, user systems 26, user

interface modules 28 and network 36. Additionally, server system 14 may be connected to a server cache 44 and network output module 22 may be connected to a network output module cache 46. Further, a network server 42 may be provided to generate output from network output module 22 across network 36 to user systems 26.

5 In system 50, network 36 may comprise an intranet, the Internet or any other network system that enables communication between a server and user system such as servers 14 and 42 and user systems 26. Network server 42 may comprise a web server and user system 26 may comprise computers connected over the Internet and World Wide Web to web server 42.

10 Server cache 44 may be provided to store calculated report results in a relational database structure. By providing a server cache 44 with server system 14, complex business reports may be accessed with one simple server request statement rather than having to access data warehouse 12 each time that a complex business report is requested. Additionally, through the use of server cache 44, load balancing
15 may be provided. Further, for security purposes, server cache 44 may be used to index result tables by filter/template combination and by user.

Network output module cache 46 may be provided to store dynamic web pages created to display report results in files through network server 42. Cache results in network output module 46 may retain links to data warehouse content for *ad hoc*
20 reporting and drilling.

Various security level architectures may be employed, although not depicted in Fig. 5. For example, a database level security may be provided between data warehouse 12 and server system 14. A network security user validation module may

be provided between network output module 22 and network server 42. Further, firewall and data encryption techniques may be provided within network 36 to prevent access to network server 42 from unqualified or unauthorized user systems 26. Also, an API may be provided to enable creation of further security level clearances to prevent user systems 26 access to data and data warehouse 12 for server cache 44 to which they do not have authorization.

Operation of system 50 may now be described. System 50 may enable users to request reports from server system 14 to be processed against data warehouse 12 through the use of a user interface module, such as a web browser operatively connected over the Internet to a network output module 22. As such a report may be requested by a user through user interface module 28 of user system 26.

To access the data warehouse, a user may first log in to network server 42 via user system 26. The user may do so by selecting a particular web site maintained by network server 42 or some other mechanism may be used. Network server 42 then presents the user with a view using user interface module 28 (*e.g.*, a web browser) comprising one or more options. A view may comprise a page or series of pages from a website, for example. One option presented in the view may be a mechanism that enables a user to generate a report request through this mechanism in the view. User interface module 28 may be used to submit a request for a report selected by the user to network server 42 over network 36. Network server 42 communicates the request to network output module 22. Network output module 22 receives the request and formulates a report request to be sent to server system 14 for processing using data warehouse 12. Network output module 22 may also perform other steps, as described

in detail below with respect to Fig. 3. Network output module 22 may also cooperate with an agent module operatively connected over system 50 to formulate a report. Server system 14 then accesses data warehouse 12 to perform all data retrieval and calculations to provide the report.

5 Network output module 22 may provide the functionality to determine whether a report request is the same or substantially the same as a previously requested report prior to forwarding a report to server system 14 for processing. Upon receipt of a report request, network output module 22 also controls network server 42 to present a new view to the user through user interface module 28 via network 36 while
10 continuing to monitor progress of submitted reports identified for the particular user. This enables users to perform other tasks and thereby provides the asynchronous nature of the present invention. For example, the user may operate other views, pages, sites or any other action using user interface module 28, without having to wait for the report generation to be complete before being able to do so.

15 After the report generation is complete, server system 14 communicates the report to network output module 22. Network output module 22 prepares the report for presenting to the user(s) that requested the report using user interface module 28. As such, network output module 22 controls asynchronous processing of reports requested by the users.

20 Fig. 6 illustrates various components and objects of system 50 that perform tasks and functions described above. These objects may reside anywhere within system 50, but the location of each according to one embodiment is described as well. System 50 comprises a report requesting object 112, a report request receiving object

114, a report request submitting object 116, an option presenting object 118, a selected option receiving object 120, and a selected option submitting object 122. Report requesting object 112 communicates with report request receiving object 114 to enable a user to request a report to be generated. Report requesting object 112 may
5 comprise functionality provided in a web page presented through user interface module 28. Report request receiving object 114 may comprise a module that is a portion of network output module 22. Report request submitting object 116 submits the user's report request for generation. Report request submitting object 116 may comprise an object that is a portion of network output module 22.

10 After a request has been submitted, option presenting object 118 may present various options for the report. Selected option receiving object 120 communicates with option presenting object 118 to select the option(s) selected by the user. Options may include specification of the name, content, project, filter, template and other components of a report as may be used by server system 14. Selected option
15 submitting object 122 then submits the options selected by the user for processing. A report generating object 124 generates the requested report asynchronously based on the options selected by the system user. A report list creating object 126 creates a report which lists one or more reports requested. If a request for a report that has already been requested, network output module 22 adds the request, including an
20 identification of the system user requesting the report, to the report entry without generating another report. A status page creating object 128 creates a status page which identifies the status of one or more requested reports. The status page may include options which may be selected by the system user (e.g., cancel request).

Option presenting object 118 and selected option submitting object 122 may comprise functionality presented in a page to the user via user interface module 28. Other objects described may comprise components of network output module 22.

Through an administrator module within network output module 22, an administrator may also monitor performances of each report, filter reports by pending, error and completed requests, cancel or modify reports, and may resubmit a report if it creates an error. The administrator module may have restricted access to prevent users from canceling other user requests. The administrator module may also allow for setting global settings for the system. Further, the administrator module may also enable administrators to view history lists, status lists, and cancellation pages from the viewpoint of users on the system. This enables administrators to be able to see what the user is seeing and therefore, troubleshoot or answer questions about the performance and operation of the system.

As described above, therefore, the present invention provides the ability for users and system administrators to cancel report requests. If a user desires to cancel a report request, the user may submit a report request cancellation by, for example, clicking on a "cancel" button using a conventional computer mouse or by selecting "cancel" from a standard pull-down menu. After a report request cancellation has been submitted, the system removes the request for that user from the report entry in the report list and then determines whether any other requests by other users remain for that report entry. If the report entry contains additional requests for that report, the system continues generating the report. This method enables users to cancel respective report requests while continuing report generation for other system users.

Additionally, this reduces system overhead by generating a single report for multiple requests. For example, if several users request the same report, only one report will be generated unlike prior systems where reports are generated for each request submitted by a user. For example, if ten users submitted a single request for the same report, ten
5 reports would be generated. The invention, however, generates only one report that is used for each of the requests submitted by the ten (10) system users.

Another feature of the invention enables notification to a system user of a completed report. Once a report is generated, the system notifies the user that requested the report via e-mail, telephone, fax, pager, or other method that the report
10 is completed. This may be done using known broadcast technology. For example, the invention may use Microstrategy's DSS Broadcaster™ to implement this notification feature.

Users may request and format reports generated using standard spreadsheet software using a network interface device (*e.g.*, a web browser). The web browser
15 may be, for example, Microsoft Internet Explorer™, Netscape Navigator™, etc. Use of a web browser allows users to easily manipulate and generate reports without requiring the user to have a specific application loaded on the user system.

The present invention may be used in an overall system comprising a plurality of other systems for OLAP and decision support. According to one embodiment of the
20 invention, a decision support system 10 may be provided as depicted in Fig. 7. System 10 may comprise a data warehouse 12, a server system 14, an architect module 16, an administrator module 18, a broadcast module 20, a network output module 22, a plurality of user systems 26, and an object creation module 24. Data warehouse 12

may comprise any data warehouse or data mart as is known in the art, including a relational database management system ("RDBMS") or a multidimensional database management system ("MDBMS") or a hybrid of the two. As discussed above, data warehouse 12 consolidates source system transaction data, organizes that data by business subject areas called dimensions and often includes data history for long term periods, such as two to five years. A data mart includes dimensions which tend toward application-specific subjects instead of integrating the entire business in one database such as departmental analysis data, data mining, and mass mailing lists.

Server system 14 comprises an OLAP server system for accessing and managing data stored in data warehouse 12. Server system 14 may comprise a ROLAP engine, MOLAP engine or a HOLAP engine according to different embodiments. Specifically, server system 14 may comprise a multithreaded server for performing sophisticated analysis directly against data warehouse 12. One embodiment of server system 14 may comprise a ROLAP engine known as DSS Server™ offered by MicroStrategy.

Architect module 16 may comprise a module that enables developers to create and maintain data and metadata in data warehouse 12. Metadata may be considered to be data about data, such as data element descriptions, data type descriptions, attributes/property descriptions, range/domain descriptions, and process/method descriptions. Data and metadata stored in data warehouse 12 may thus be modified and organized by architect module 16. According to one embodiment of the invention, architect module 16 may comprise a software package known as DSS Architect™ offered by MicroStrategy.

Administrator module 18 may comprise a module for facilitating the development, deployment and management of data warehouse applications supporting large volumes of users over various distribution mechanisms. Administrator module 18 may comprise an object manager and a warehouse monitor. The object manager
5 allows objects to be shared across databases for easy migration from development to production. The warehouse monitor provides performance monitoring and management tools to support thousands of users across a distributive database environment. The warehouse monitor collects statistics for the purpose of identifying performance bottlenecks, warehouse tuning, and cost analysis. According to one
10 embodiment of the invention, administrator module 18 may comprise a module known as DSS Administrator™ offered by MicroStrategy.

Server system 14 also connects to an object creation module 24. Object creation module 24 may comprise a component object model (“COM”) application program interface (“API”) for custom decision support development. According to
15 one embodiment of the invention, object creation module 24 may comprise a software module known as DSS Objects™ offered by MicroStrategy. Additionally, custom applications may interface with object creation module 24 including Delphi, Visual Basic, and C++ programming modules.

Server system 14 also interfaces with a number of devices that enable users to
20 access the contents of the data warehouse to perform various data analyses. One embodiment of such a system comprises a user system 26 with an agent module 28 operating thereon. Agent module 28 may comprise a decision support interface to

enable a user to provide integrated query and reporting, powerful analytics, and decision support through any standard user system 26.

Agent module 28 may operate on any user system including personal computers, network workstations, laptop computers or any other electronic device
5 connected to server system 14. Agent module 28 may enable the user to define queries to be performed against the data contained in data warehouse 12 using components, templates, filters, reports, agents, etc. Components may include dimensions, attributes, attribute elements, and metrics; in other words, the building blocks for templates, filters, and reports. Templates generally define a report format
10 and specify the attributes, dimensions, metrics, and display properties comprising a report. Filters generally qualify report content and identify the subset of data warehouse 12 to be included in a report. Reports are generally understood to be a data analysis created by combining a template (the format) with a filter (the content). Agents are a group of reports cached on a time -or event- based schedule for rapid
15 retrieval and batch processing.

Accordingly, agent module 28 enables a user access to the contents of data warehouse 12 to provide detailed analysis on an *ad hoc* basis. According to one embodiment of the invention, agent module 28 may comprise a software package known as DSS Agent™ offered by MicroStrategy. One of the advantages of DSS
20 Agent™ includes its use of a ROLAP architecture on server system 14 and a RDBMS in data warehouse 12 to provide a more scaleable environment. Through DSS Agent™, a user can “drill down” which allows the user to dynamically change the level of detail in a report. Drilling down allows the user to drill to a lower level

attribute so that the resulting report displays data with a greater level of detail. For example, one can drill down from year to month to week to day. DSS Agent™ also enables users to “drill up” to a higher level attribute. Drilling up summarizes the selected data to a higher level total. For example, one can drill from day to week to
5 month to year. DSS Agent™ also enables a user to “drill within.” Drilling within allows a user to go to a different hierarchy within the same dimension. Drilling within is often used to examine the characteristics of selected data. For example, drilling within enables a user to drill from item to color when looking at a particular retail item such as an automobile, clothing or the like. Drilling across allows the user to
10 drill to an altogether different dimension or subject area. For example, one can drill across from a region to a month. Accordingly, through use of agent module 28, server system 14, and data warehouse 12, drilling is a powerful tool that is easily implemented using a ROLAP architecture which is not as easily accessible in MOLAP unless a user also implements the complicated structure of a HOLAP architecture.

15 Along with agent module 28, user system 26 may also include a report writing module 30, an executive module 32, and a spreadsheet module 34. Report writing module 26 may comprise an OLAP report writer. Executive module 32 may comprise a module design tool for developing custom EIS applications. This module is a design tool for developing briefing books that provide high level users with a series of
20 views that describe their business. Once created, end users can access briefing books through agent module 28 in EIS mode. Such a system is easily implemented with agent module 28 by compiling sets of analyses into dynamic pages that immediately

focus users on their key business drivers. One embodiment of executive module 32 may comprise the software made DSS Executive™ offered by MicroStrategy.

Spreadsheet module 34 may comprise an add on to existing spreadsheet programs or may comprise an entirely new spreadsheet program. Spreadsheet module
5 34 may enable reports and analyses generated from agent module 28 to be presented in a traditional spreadsheet program format to enable users to view results in preexisting front end interfaces. Spreadsheet module 34 may comprise the Microsoft Excel™ spreadsheet program offered by Microsoft and an Excel™ Add-In program offered by MicroStrategy.

10 Another module for accessing content of server system 14 may comprise a network output module 22. Network output module 22 offers the advantage of enabling user system 26 access to server system 14 in data warehouse 12 without requiring an additional agent module 28 to be stored on user system 26. Instead, user system 26 may have a user interface module 38 residing on user system 26. User
15 interface module 38 may comprise any module that enables a user system to interface with the network output module over a network 36. According to one embodiment of the invention, network 36 may comprise an intranet or the Internet or other developed Internet-type networks. Further, user interface module 38 may comprise any standard browser module such as Microsoft Internet Explorer™ or Netscape Navigator™. As
20 most user systems 26 already have a user interface module 38 stored and operating thereon, network output module 22 offers the advantage of enabling users access to server system 14 in data warehouse 12 without learning to operate a new module such as agent module 28. One embodiment of network output module 22 may comprise a

web-based module called DSS Web™ offered by MicroStrategy. Accordingly, in one embodiment, a user can access server system 14 through a standard web browser, such as Microsoft Internet Explorer™, over the Internet through network output module 22, such as DSS Web™.

5 In this embodiment, network output module 22 may comprise a World Wide Web tool used in conjunction with server system 14 for allowing users to deploy data warehouse/decision support applications over the Internet using industry standard World Wide Web browsers as a client. As a result, a user can access the data warehouse with little or no client maintenance, little or no software to install, and only
10 a small amount of additional training while still maintaining all of the capabilities of agent module 28. One embodiment of such a network output module 22 comprises DSS Web™ offered by MicroStrategy. Such an embodiment provides a broad array of options for viewing information sets, such as spreadsheet grids and a wide variety of graphs. Through this module's exceptional reporting capabilities, users receive key
15 elements of a report in easily interpretable, plain messages. This module also allows users to "drill" dynamically to a lower level of detail to view the underlying information or to create and save new analyses. For sensitive information, this module provides security plug-ins that allow the user to extend the standard security functionality with additional user authentication routines. This module may also
20 provide an interface API that allows users to customize, integrate, and imbed this functionality into other applications. For example, the data syndicator for health care information could utilize this module with a customize interface to sell access to

health care information to Health Maintenance Organizations, hospitals, and pharmacies.

Another module that enables user access to the data warehouse 12 comprises a broadcast module 20. Broadcast module 20 may comprise an information broadcast
5 server designed to deliver personalized messages to multiple recipients via user devices 40. User devices 40 may comprise electronic mail, facsimile, pager, mobile phone, telephone, and multiple other types of user information devices. Broadcast module 20, like agent module 28 and network output module 22, enables users to define queries and reports that are to be run against server system 14 and data
10 warehouse 12. Broadcast module 20 may then send personalized information to users at predefined intervals or when criteria specified in their reports exceed predefined thresholds. Broadcast module 20 may continually monitor and run such reports to enable users to receive appropriate information as it becomes available. Broadcast module 20 may also personalize the report to ensure that users see only that portion of
15 a report that is relative to that user and thereby frees analysts from acting as the information provider. Broadcast module 20 may also provide a wide range of content formatting options to enable users to receive information in a form that they can best understand. One such formatting may include formatting a request for a pager, facsimile, mobile phone, electronic mail, web or other output user device 40.

20 Broadcast module 20 may specify a mechanism called a service for generating information and a subscription that determines which users receive information from which service. Accordingly, a service may be tied to, for example, a specific report, series of reports or elements within a report, etc. Also, subscribers may include users,

groups of users or only specific user devices 40 for a particular user. Broadcast module 20 may output information to user devices 40 using push technology. In other words, as soon as a criteria is established, broadcast module 20 generates a report to a user device 40. Accordingly, a user can thus have up-to-date information about the contents of data warehouse 12 without having to submit a query or log-in to a software module such as agent module 28 or network output module 22 to initiate the query.

In such an overall system, the present invention may enable asynchronous report generation through operation of network output module system 22 as described above. Other embodiments and uses of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. The specification and examples should be considered exemplary only. The scope of the invention is only limited by the claims appended hereto.

CLAIMS

What is claimed is:

1. A network-based system for enabling users connected over the network to an OLAP system to select formatting options for reports requested for processing
5 by the OLAP system comprising:

report requesting means for enabling a user to select a workbook comprising one or more reports to be requested for processing by the OLAP system; and

format specification means for enabling the user to select the format for each report in the workbook.
- 10 2. The network-based system of claim 1 wherein the format specification means enables the user to specify multiple levels of formats to be applied to the reports in the workbook.
3. The network-based system of claim 2 wherein the multiple levels of formats are applied in a hierarchical manner so that lower level formats supersede
15 higher lever formats.
4. The network-based system of claim 3 wherein the levels of formats comprise global formats, report formats, attribute/element/metric formats, and value formats in that hierarchical order.
5. The network-based system of claim 1 wherein the format of one of the
20 workbooks comprises a merged report format.
6. The network-based system of claim 1 wherein the format of one of the reports comprises a combined grid and graph.

7. The network-based system of claim 1 wherein the format comprises rules to be applied to the report that the user may specify.

8. The network-based system of claim 1 wherein the format of a report comprises an autoformat that enables each recipient of the report to specify the format in which that recipient views the report when the report is received.

9. The network-based system of claim 1 wherein the format of a report comprises a scripted format created by a user for specialized formatting of the report.

10. A method for enabling users connected over the network to an OLAP system to select formatting options for reports requested for processing by the OLAP system comprising the steps of:

enabling a user to select a workbook comprising one or more reports to be requested for processing by the OLAP system; and

enabling the user to select the format for each report in the workbook.

11. The method of claim 10 further comprising the step of enabling the user to specify multiple levels of formats to be applied to the reports in the workbook.

12. The method of claim 11 wherein the multiple levels of formats are to be applied in a hierarchical manner so that lower level formats supersede higher level formats.

13. The method of claim 12 wherein the levels of formats comprise global formats, report formats, attribute/element/metric formats, and value formats in that hierarchical order.

14. The method of claim 10 wherein the format of one of the workbooks comprises a merged report format.

15. The method of claim 10 wherein the format of one of the reports comprises a combined grid and graph.

16. The method of claim 10 wherein the format comprises rules to be applied to the report that the user may specify.

5 17. The method of claim 10 further comprising the step of enabling the user to specify the format of a report to be an autoformat that enables each recipient of the report to specify the format in which that recipient views the report when the report is received.

10 18. An electronic storage medium having code embodied therein for causing a processing system to enable users connected over a network to specify format of reports requested from an OLAP system, the medium comprising:

code for causing a processor to enable a user to select a workbook comprising one or more reports to be requested for processing by the OLAP system; and

15 code for causing a processor to enable a user to select the format for each report in the workbook.

19. The medium of claim 18 further comprising code for causing a processor to enable the user to specify multiple levels of formats to be applied to the reports in the workbook.

20 20. The method of claim 19 wherein the multiple levels of formats are to be applied in a hierarchical manner so that lower level formats supersede higher lever formats.

21. The medium of claim 18 further comprising code for causing a processor to select a merged report format.

22. The medium of claim 18 further comprising code for causing a processor to enable a user to select the format of a report to include a combined grid and graph.

23. The medium of claim 18 further comprising code for causing a
5 processor to enable a user to select the format of a report to be an autoformat that enables each recipient of the report to specify the format in which that recipient views the report when the report is received.

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2

SYSTEM AND METHOD FOR NETWORK USER INTERFACE OLAP

REPORT FORMATTING

Abstract of the Disclosure

A network-based system for enabling users connected over the network to an OLAP system to select formatting options for reports requested for processing by the OLAP system. The system enables a user to select a workbook comprising one or more reports to be requested for processing by the OLAP system or to select a stand-alone report. The system then enables a user to select the format for each report in the workbook or to a stand-alone report. The format specified may be multiple levels of format that are applied in a hierarchical manner including global formats that apply to all reports in a workbook, report formats which apply to specific reports in the workbooks, attribute/element/metric formats which apply to specific attributes, elements, and metrics of a report, and value formats which apply to specific values within an element of a report. The formats specified may also comprise merged report formats that specify the merger of two or more reports, combined grid and graphs within a report and an autoformat that enables each recipient of the report to specify the format in which that recipient views the report when the report is received. Formatting may be auto-formats or through scripting

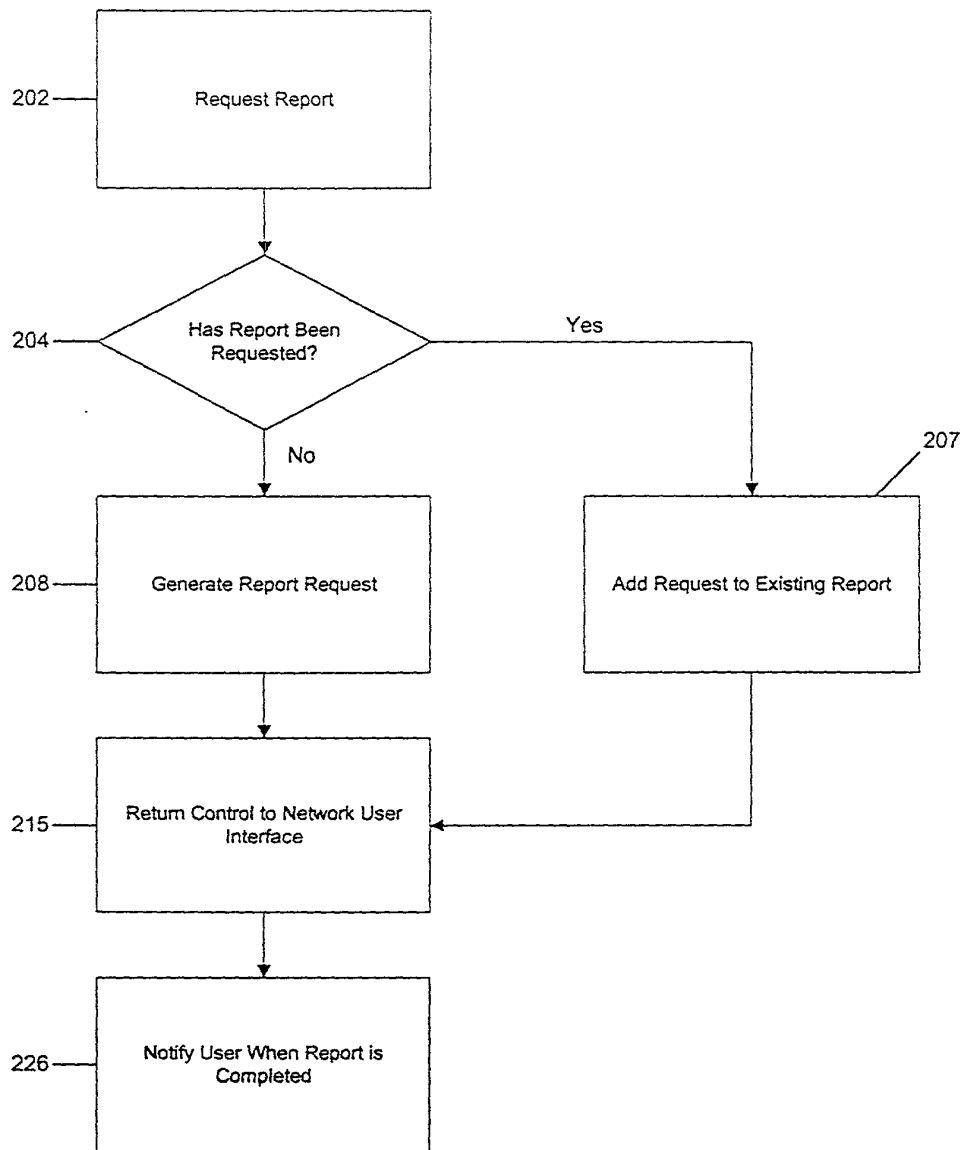


Figure 1

200

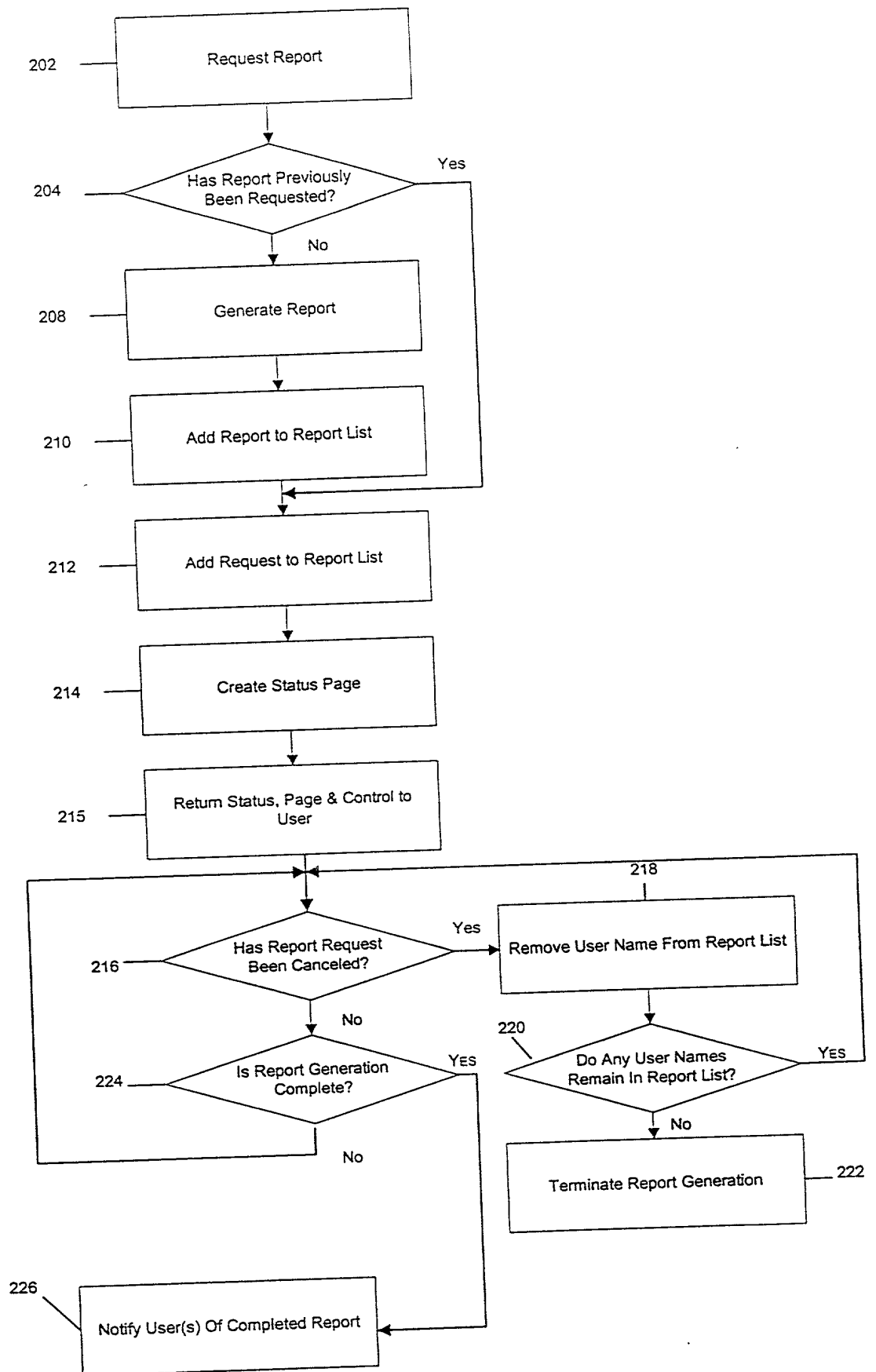


Figure 2

	302	303	304	306
	Name	Folder	Sumbitted ▼	Details
✿	Weekly Churn Report for . . .	Churn Reports	9/30/98 5:27 pm	Executing SQL
✿	<u>Regional Sales</u>	Sales R	This is a description of a report. It appears after the cursor has paused over the selected report for > 1 second.	
✿	<u>Auto Prompt Report (1)</u>	Auto Prompt Reports	9/30/98 5:20 pm	Ready
	Step 1 (Geography) [Region] Northeast NOT Boston Step 2 (Time) [Year] 1996, 1997, 1998 [Month] NOT January Step 3 (Product) [Athletic] Jordan BBall, Nike Cap, Reebok Running Shoes [Color] Black, White, Red, Yello, Green, Blue			
★	My New Report	My Reports	9/30/98 5:18 pm	Error Executing SQL
✿	<u>Sales to Date</u>	Sales Reports	9/30/98 5:01 pm	Ready

300

Fig. 3

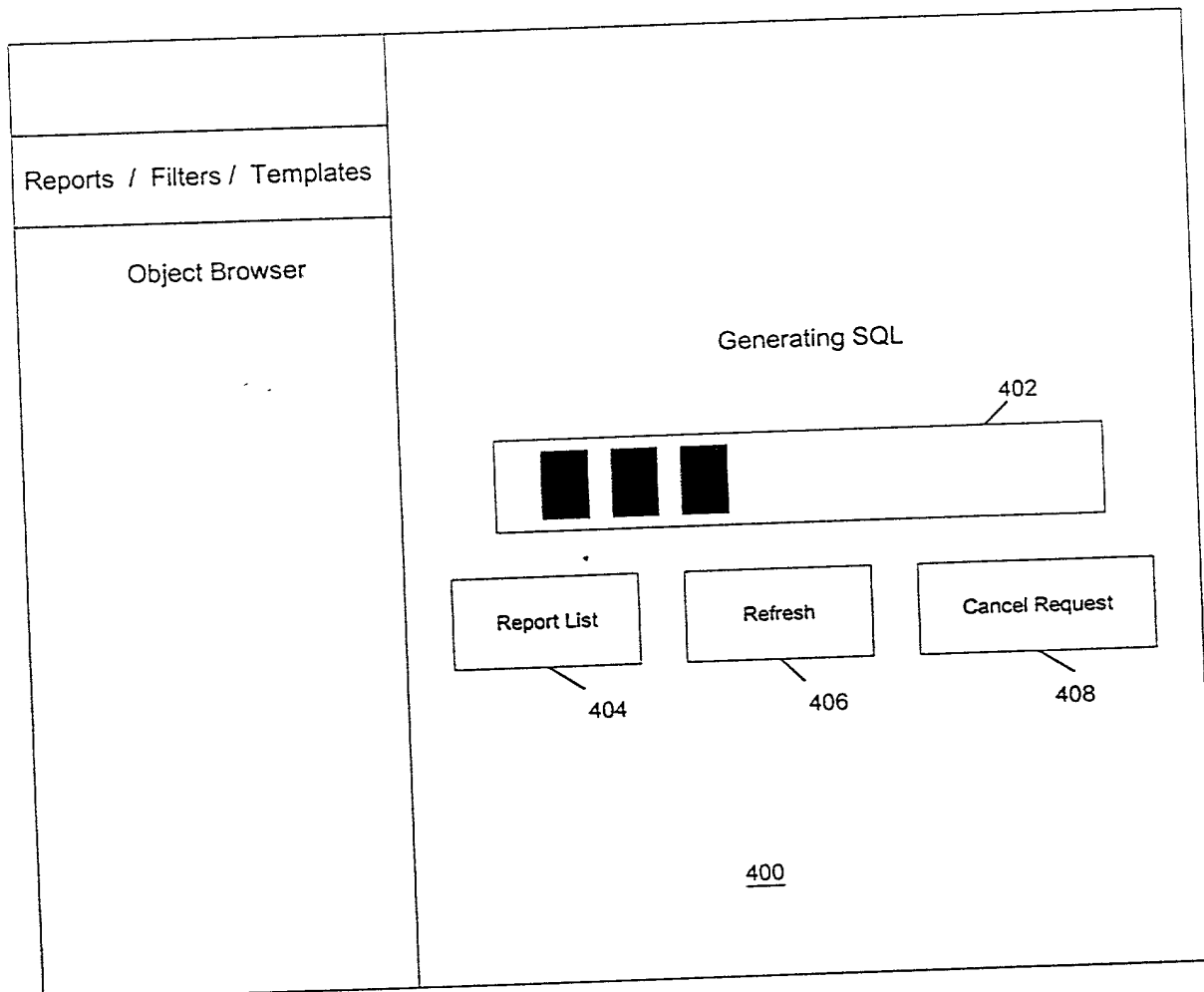


Figure 4

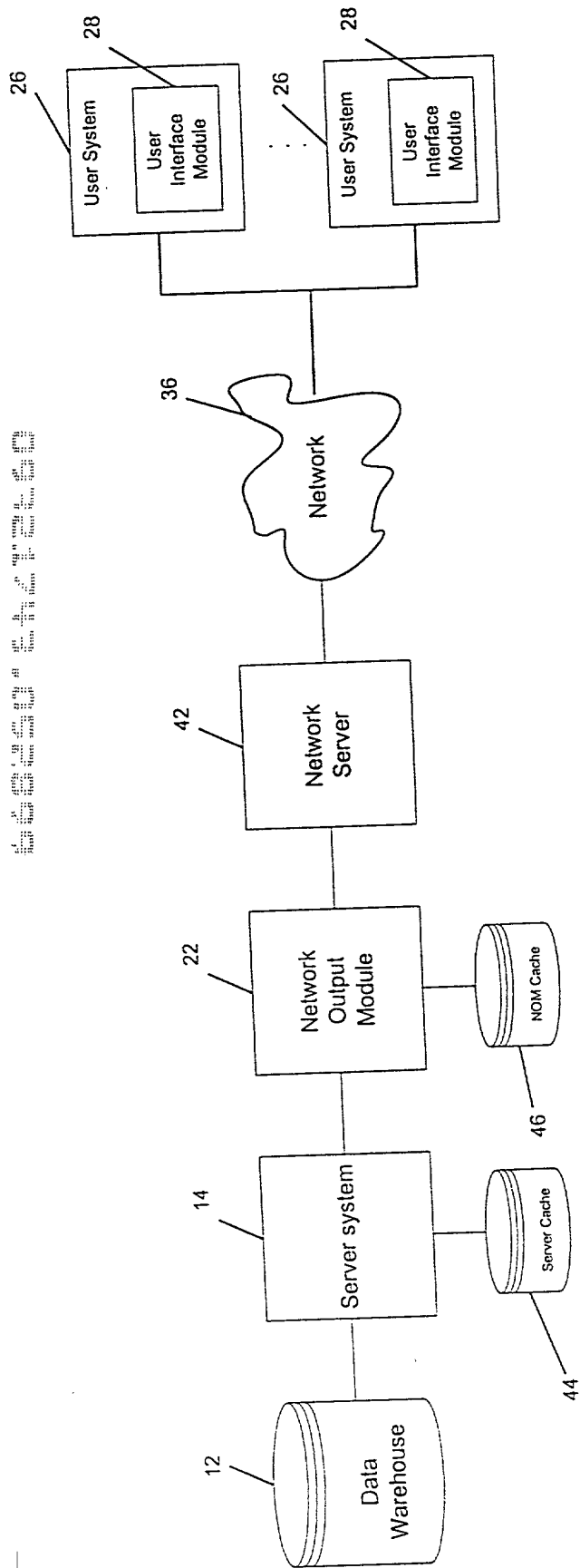


Figure 5

50

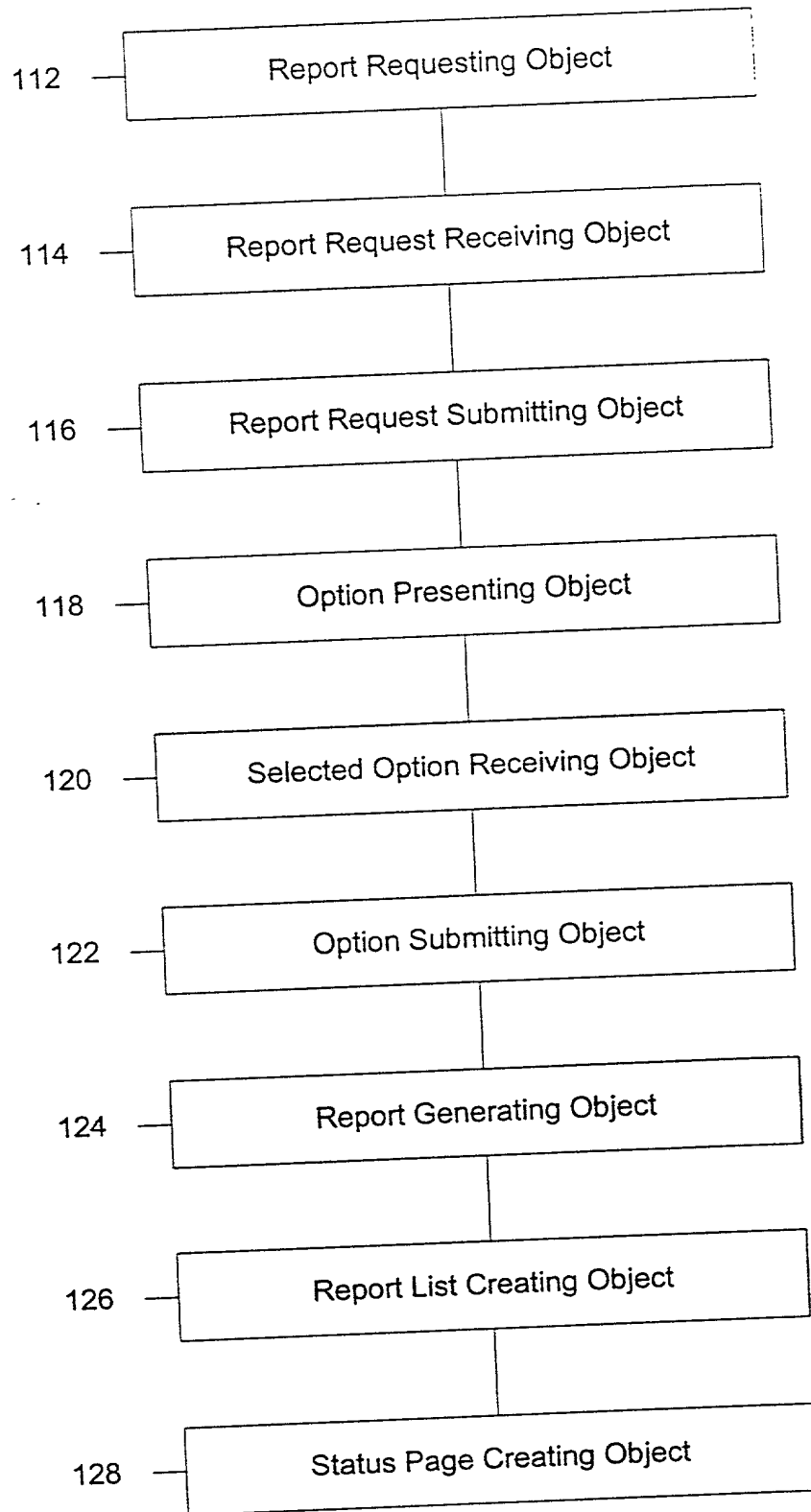


Figure 6

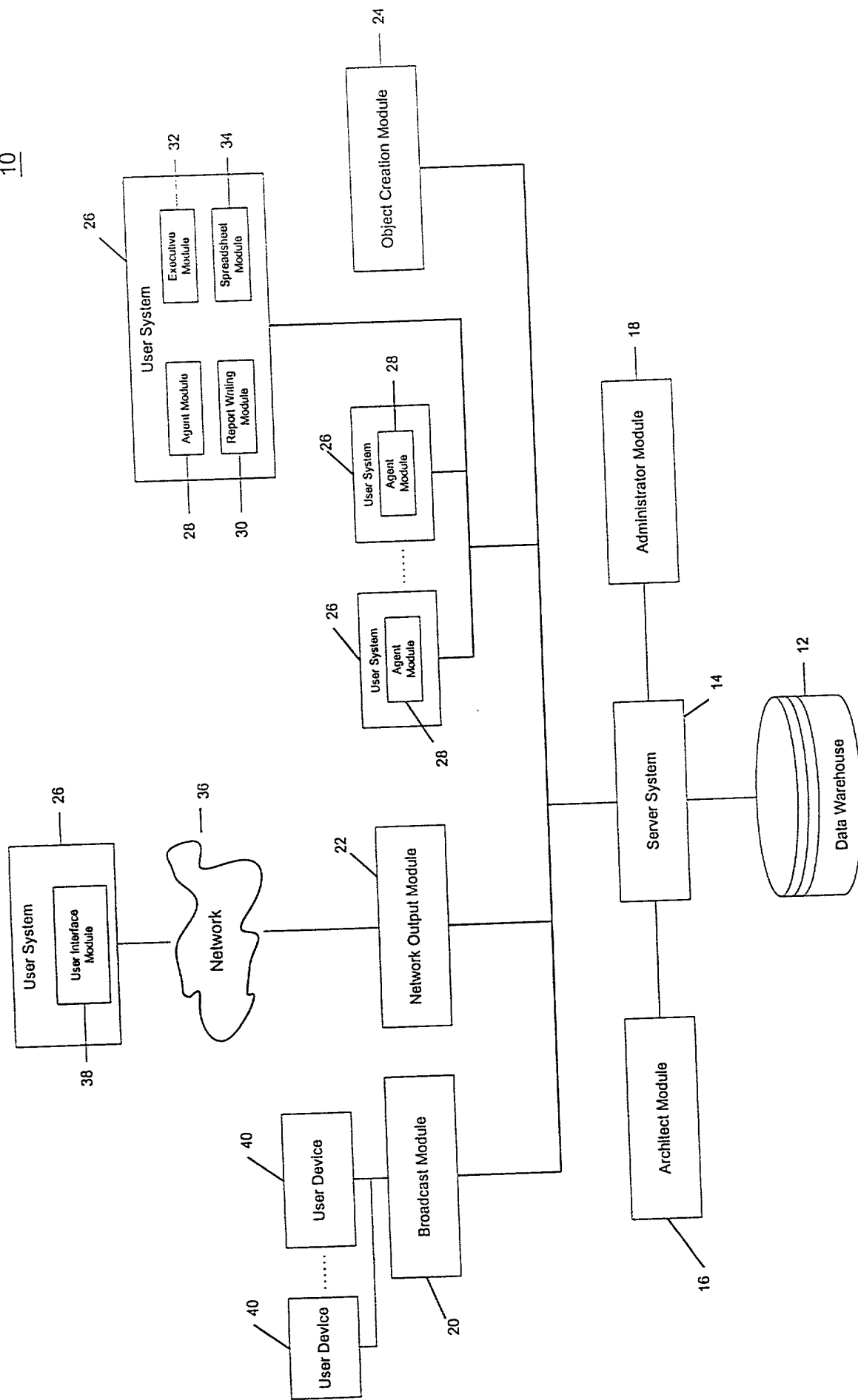


Figure 7

JOINT DECLARATION FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenship are as stated below next to our names;

We believe that we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled SYSTEM AND METHOD FOR NETWORK USER INTERFACE OLAP REPORT FORMATTING, the specification of which

☒ is attached hereto.

☐ was filed on _____ as Application Serial Number _____ and was
amended on _____
(if applicable)

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to in this declaration.

We acknowledge the duty to disclose all information known to us to be material to the patentability of this application, as defined in 37 C.F.R. § 1.56.

We acknowledge the duty to disclose to the Office all information known to us to be material to patentability as defined in § 1.56, which became available between the filing date of the prior application and the national or PCT international filing date of the continuation-in-part application.

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119
				Yes <input type="checkbox"/> No <input type="checkbox"/>
				Yes <input type="checkbox"/> No <input type="checkbox"/>
				Yes <input type="checkbox"/> No <input type="checkbox"/>
				Yes <input type="checkbox"/> No <input type="checkbox"/>

Prior United States Application(s)

We hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, we acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial Number	Date of Filing (day, month, year)	Status - Patented, Pending, Abandoned

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith the following attorneys, their registration numbers being listed after their names:

Thomas J. Scott, Jr., Registration No. 27,836; Stanislaus Aksman, Registration No. 28,562; Kevin J. Dunleavy, Registration No. 32,024; James G. Gatto, Registration No. 32,694; Scott D. Balderston, Registration No. 35,436; Tyler S. Brown, Registration No. 36,465; Christopher C. Campbell, Registration No. 37,291; Henry C. Su, Registration No. 37,738; Brian M. Buroker, Registration No. 39,125; Thomas G. Woolston, Registration No. 40,235; Charles F. Hollis, Registration No. 40,650; Kevin T. Duncan, Registration No. 41,495; Jonathan D. Link, Registration No. 41,548; Christopher J. Cuneo, Registration No. 42,450; Stephen T. Schreiner, Registration No. 43,097; Raphael A. Valencia, Registration No. 43,216; and George B. Georgellis, Registration No. P43,632.

All correspondence and telephone communications should be addressed to Hunton & Williams, 1900 K Street, N.W., Suite 1200, Washington, D.C. 20006-1109, telephone number (202) 955-1500, which is also the address and telephone number of each of the above listed attorneys.

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Full Name of
First Inventor **HAROLD** **Lee** **C.**
Family Name First Given Name Second Given Name

Residence **6705G Washington Blvd., Arlington, VA 22213**

Citizenship **United States**

Post Office
Address **Same as above**

Signature _____ Date _____

Full Name of
Second Inventor **ZHU** **Qing** **Second Given Name**
Family Name First Given Name

Residence **13150 Copper Brook Way, Herndon, Virginia 20171**

Citizenship **China**

Post Office
Address **Same as above**

Signature _____ Date _____

Full Name of
Third Inventor **LIEBL** **Herbert** **Second Given Name**
Family Name First Given Name

Residence **2228 Mohegan Drive #302, Falls Church, Virginia 22043**

Citizenship **Austria**

Post Office
Address **Same as above**

Signature _____ Date _____

HUNTON & WILLIAMS
1900 K Street, N.W., Suite 1200
Washington, D.C. 20006-1109
Tel: (202) 955-1500